In the claims:

1. (currently amended): A method of software pipelining program loops having irregular loop control comprises the steps of:

determining which instructions in loop code in a memory may be speculatively executed ~~without special hardware support or special loop control instruction~~,

storing in a computer memory a set of registers that are modified by an instruction and are alive out of the loop, and

modifying the program code so that the values of those registers are saved to a temporary register during all proper iterations, and

copying back to a register the value of the temporary register once the loop is completed.

2. (original): The method of Claim 1, applied to any program loops wherein a minimum trip count is reduced to one.

3. (original): The method of Claim 1, applied to any program loops wherein a need for a multiversion code is eliminated.

4. (previously amended): A method for software pipelining of irregular conditional control loops includes pre-processing the loops so they can be safely software pipelined, comprising the steps of:

pre-processing each instruction in the loop in turn;

if the instruction can be safely speculatively executed, leaving the instruction alone;

if it could be safely speculatively executed except that it modifies registers that are live out of the loop, pre-processing the instruction using register copying and otherwise using predication.

5. (cancelled) The method of Claim 4, including the step of pre-processing the instruction by predication if it could not be safely speculatively executed.

6. (cancelled) The method of Claim 4, wherein if it could be safely speculatively executed except that it modifies registers that are live out of the loop, pre-processing the instruction by register copying.